

## 6

# Objective Analysis (*little\_r*)

---

## 6.1 Purpose of Objective Analysis

The goal of objective analysis in meteorological modeling is to improve meteorological analyses (the *first guess*) on the mesoscale grid by incorporating information from observations. Traditionally, these observations have been “direct” observations of temperature, humidity, and wind from surface and radiosonde reports. As remote sensing techniques come of age, more and more “indirect” observations are available for researchers and operational modelers. Effective use of these indirect observations for objective analysis is not a trivial task. Methods commonly employed for indirect observations include three-dimensional or four-dimensional variational techniques (“3DVAR” and “4DVAR”, respectively), which are used for direct observations as well.

The MM5 system has long included packages for objective analysis of direct observations: the venerable RAWINS program and the upstart LITTLE\_R program. A recent addition to the MM5 system is the 3DVAR package, which allows for variational assimilation of both direct observations and certain types of indirect observations. This chapter discusses the objective analysis program, LITTLE\_R, which is perhaps best suited to new MM5 users. Some reference is made to the older RAWINS program (some details are available in Appendix F). Discussion of 3DVAR is reserved for Appendix E.

The analyses input to LITTLE\_R and RAWINS as the first guess are generally the low-resolution analyses output from program REGRID. LITTLE\_R and RAWINS may also use an MM5 forecast (through a back-end interpolation from sigma to pressure levels) as the first guess.

LITTLE\_R and RAWINS capabilities include:

- Choice of Cressman-style or Multiquadric objective analysis.
- Various tests to screen the data for suspect observations.
- Procedures to input bogus data.
- Expanded Grid: If you used an expanded grid in TERRAIN and REGRID, the objective analysis can incorporate data from outside your grid to improve analyses near the boundaries. These programs cut down the expanded grid to the unexpanded dimensions on output.

Output from the objective analysis programs is used to:

- Provide fields for Initial and Boundary conditions (through program INTERPF).
- Provide 3-d fields for analysis-nudging FDDA (through program INTERPF).
- Provide surface fields for surface-analysis-nudging FDDA.

## 6.2 RAWINS or LITTLE\_R?

Users are strongly encouraged to use LITTLE\_R for the objective analysis step. Most of what you'll need to do is done more easily in LITTLE\_R than in RAWINS.

## 6.3 Source of Observations

Input of observations is perhaps the greatest difference between LITTLE\_R and RAWINS. RAWINS was developed around a specific set of data in a specific format. Incorporating data into RAWINS from unexpected sources or in different formats tends to be a challenge. LITTLE\_R specifies it's own format for input (which has it's own challenges), but is better suited for users to adapt their own data.

RAWINS incorporates data from NCEP operational global surface and upper-air observations subsets as archived by the Data Support Section (DSS) at NCAR.

- Upper-air data: RAOBS (ADPUPA), in NMC ON29 format.
- Surface data: NMC Surface ADP data, in NMC ON29 format.

NMC Office Note 29 can be found in many places on the World Wide Web, including:

<http://www.emc.ncep.noaa.gov/mmb/papers/keyser/on29.htm>.

LITTLE\_R reads observations provided by the user in formatted ASCII text files. The LITTLE\_R tar file includes programs for converting the above NMC ON29 files into the LITTLE\_R Observations Format. Users are responsible for converting other observations they may want to provide LITTLE\_R into this format. The details of this format are provided in section 6.12.

## 6.4 Objective Analysis techniques in LITTLE\_R and RAWINS

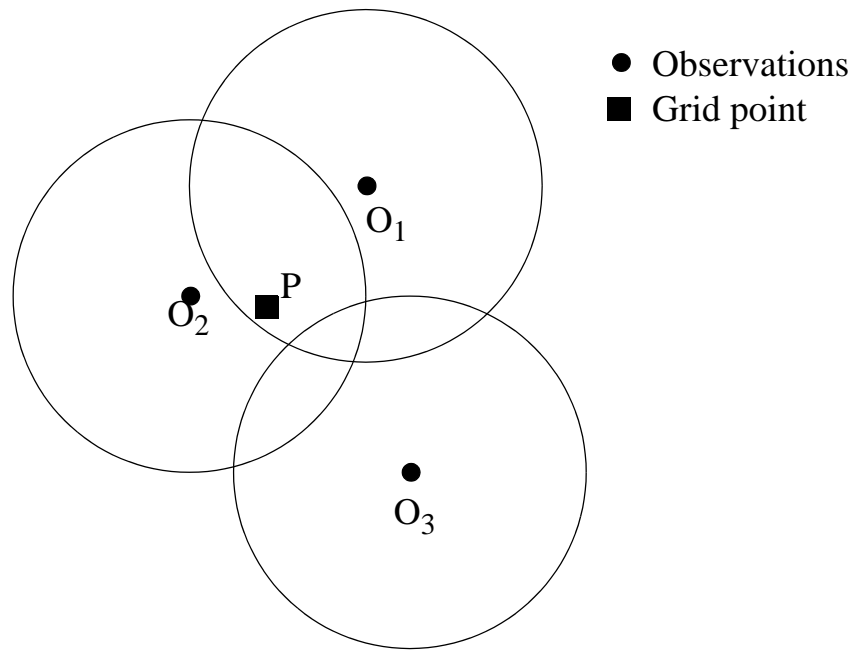
### 6.4.1 Cressman Scheme

Three of the four objective analysis techniques used in LITTLE\_R and RAWINS are based on the Cressman scheme, in which several successive scans nudge a first-guess field toward the observations.

The standard Cressman scheme assigns to each observation a circular radius of influence  $R$ . The first-guess field at each gridpoint  $P$  is adjusted by taking into account all the observations which influence  $P$ .

The differences between the first-guess field and the observations are calculated, and a distance-weighted average of these difference values is added to the value of the first-guess at  $P$ . Once all

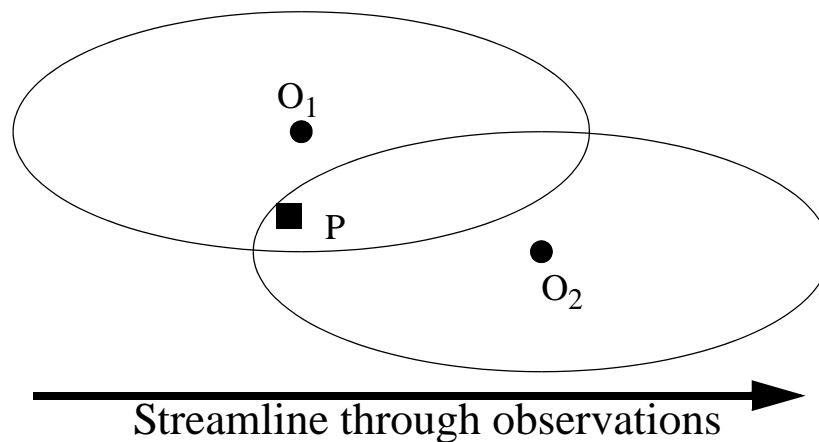
gridpoints have been adjusted, the adjusted field is used as the first guess for another adjustment cycle. Subsequent passes each use a smaller radius of influence.



Observations  $O_1$  and  $O_2$  influence grid point  $P$ ,  $O_3$  does not.

#### 6.4.2 Ellipse Scheme

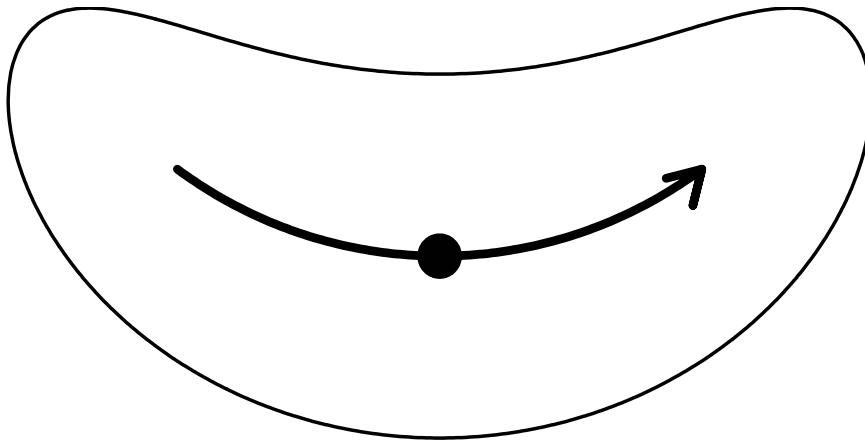
In analyses of wind and relative humidity (fields strongly deformed by the wind) at pressure levels, the circles from the standard Cressman scheme are elongated into ellipses oriented along the flow. The stronger the wind, the greater the eccentricity of the ellipses. This scheme reduces to the circular Cressman scheme under low-wind conditions.



#### 6.4.3 Banana Scheme

In analyses of wind and relative humidity at pressure levels, the circles from the standard Cress-

man scheme are elongated in the direction of the flow and curved along the streamlines. The result is a banana shape. This scheme reduces to the Ellipse scheme under straight-flow conditions, and the standard Cressman scheme under low-wind conditions.



#### 6.4.4 Multiquadric scheme

The Multiquadric scheme uses hyperboloid radial basis functions to perform the objective analysis. Details of the multiquadric technique may be found in Nuss and Titley, 1994: “Use of multiquadric interpolation for meteorological objective analysis.” *Mon. Wea. Rev.*, **122**, 1611-1631. Use this scheme with caution, as it can produce some odd results in areas where only a few observations are available.

### 6.5 Quality Control for Observations

A critical component of LITTLE\_R and RAWINS is the screening for bad observations. Many of these QC checks are done automatically in RAWINS (no user control), they are optional in LITTLE\_R.

#### 6.5.1 Quality Control on Individual Reports

Most of these QC checks are done automatically in RAWINS (no user control), most are optional in LITTLE\_R.

- Gross Error Checks (sane values, pressure decreases with height, etc.)
- Remove spikes from temperature and wind profiles.
- Adjust temperature profiles to remove superadiabatic layers.
- No comparisons to other reports or to the first-guess field.

#### 6.5.2 The ERRMAX test

The ERRMAX quality-control check is optional (but highly recommended) in both LITTLE\_R and RAWINS.

- Limited user control over data removal. The user may set thresholds which vary the tolerance of the error check.
- Observations are compared to the first-guess field.
- If the difference value (obs - first-guess) exceeds a certain threshold, the observation is discarded.
- Threshold varies depending on the field, level, and time of day.
- Works well with a good first-guess field.

### 6.5.3 The Buddy test

The Buddy check is optional (but highly recommended) in both LITTLE\_R and RAWINS.

- Limited user control over data removal. The user may set weighting factors which vary the tolerance of the error check.
- Observations are compared to both the first guess and neighboring observations.
- If the difference value of an observation (obs - first-guess) differs from the distance-weighted average of the difference values of neighboring observations, the observation is discarded.
- Works well in regions with good data density.

## 6.6 Additional Observations

Input of additional observations, or modification of existing (and erroneous) observations, can be a useful tool at the objective analysis stage.

In LITTLE\_R, additional observations are provided to the program the same way (in the same format) as standard observations. Indeed, additional observations must be in the same file as the rest of the observations. Existing (erroneous) observations can be modified easily, as the observations input format is ASCII text. Identifying an observation report as “bogus” simply means that it is assumed to be good data -- no quality control is performed for that report.

In RAWINS, the methods of adding or modifying observations are rather difficult to work with, requiring additional files with cryptic conventions. All observations provided through these files are assumed to be “good”; no quality control is performed on these observations. Don’t try this unless it’s absolutely necessary, and you’re the patient sort. However, some people actually manage to use these procedures successfully. See notes on NBOGUS, KBOGUS, NSELIM in Appendix F.

## 6.7 Surface FDDA option

The surface FDDA option creates additional analysis files for the surface only, usually with a smaller time interval between analyses (*i.e.*, more frequently) than the full upper-air analyses. The purpose of these surface analysis files is for later use in MM5 with the surface analysis nudging option. This capability is turned on by setting the namelist option F4D = .TRUE., and selecting the time interval in seconds for the surface analyses with option INTF4D.

A separate set of observations files is needed for the surface FDDA option in LITTLE\_R. These files must be listed by the namelist record2 option sfc\_obs\_filename. A separate observations file must be supplied for each analysis time from the start date to the end date at time interval INTF4D.

The LAGTEM option controls how the first-guess field is created for surface analysis files. Typically, the surface and upper-air first-guess is available at twelve-hour intervals (00 Z and 12 Z), while the surface analysis interval may be set to 3 hours (10800 seconds). So at 00 Z and 12 Z, the available surface first-guess is used. If LAGTEM is set to `.FALSE.`, the surface first-guess at other times will be temporally interpolated from the first-guess at 00 Z and 12 Z. If LAGTEM is set to `.TRUE.`, the surface first guess at other times is the objective analysis from the previous time.

## 6.8 Objective Analysis on Model Nests

LITTLE\_R and RAWINS have the capability to perform the objective analysis on a nest. This is done with a separate LITTLE\_R or RAWINS process, performed on REGRID files for the particular nest. Often, however, such a step is unnecessary; it complicates matters for the user and may introduce errors into the forecast. At other times, extra information available to the user, or extra detail that objective analysis may provide on a nest, makes objective analysis on a nest a good option.

The main reason to do objective analysis on a nest is if you have observations available on with horizontal distribution somewhat greater than the resolution of your coarse domain. There might also be circumstances in which the representation of terrain on a nest allows for better use of surface observations (i.e., the model terrain better matches the real terrain elevation of the observation).

The main problem introduced by doing objective analysis on a nest is inconsistency in initial conditions between the coarse domain and the nest. Observations that fall just outside a nest will be used in the analysis of the coarse domain, but discarded in the analysis of the nest. With different observations used right at a nest boundary, one can get very different analyses.

## 6.9 How to run LITTLE\_R

### 6.9.1 Get the source code

The source code is available via anonymous ftp, at  
`ftp://ftp.ucar.edu/mesouser/MM5V3/LITTLE_R.TAR.gz`

Download this file to your local machine, uncompress it, and untar it:

```
gzip -cd LITTLE_R.TAR.gz
tar -xvf LITTLE_R.TAR
```

You should now have a directory called LITTLE\_R. Change to that directory:

```
cd LITTLE_R
```

### 6.9.2 Generate the executable

The LITTLE\_R executable is generated through the Make utility. For a variety of platforms and architectures, the Makefile is already set up to build the executable. Simply type:

```
make
```

### 6.9.3 Prepare the observations files

For the tutorial exercises, there are prepared observations files for you to use. See the notes on the assignment.

A program is available for users with access to NCAR's computers to download archived observations and reformat them into the LITTLE\_R Observations Format. See the information about the "fetch.deck" program, in section 6.14.

A program is also available for reformatting observations from the GTS stream.

For other sources of data, the user is responsible for putting data into the LITTLE\_R Observations Format. Hence the detailed discussion of the observations format in Section 6.12.

In general, there are two overall strategies for organizing observations into observations files. The easiest strategy is to simply put all observations into a single file. The second strategy, which saves some processing time by LITTLE\_R, is to sort observations into separate files by time.

### 6.9.4 Edit the namelist for your specific case

For details about the namelist, see section 6.13. The most critical information you'll be changing most often are the start and end dates and file names. Pay particularly careful attention to the file name settings. Mistakes in observations file names can go unnoticed because LITTLE\_R will happily process the wrong files, and if there are no data in the (mis-specified) file for a particular time, LITTLE\_R will happily provide you with an analysis of no observations.

### 6.9.5 Run the program

Run the program by invoking the command:

```
little_r >! print.out
```

The ">! print.out" part of that command simply redirects printout into a file called "print.out".

### 6.9.6 Check your output

Examine the "print.out" file for error messages or warning messages. The program should have created the file called "LITTLE\_R\_DOMAIN<#>", according to the domain number. Additional output files containing information about observations found and used and discarded will probably be created, as well.

Important things to check include the number of observations found for your objective analysis, and the number of observations used at various levels. This can alert you to possible problems in specifying observations files or time intervals. This information is included in the printout file.

You may also want to experiment with a couple of simple plot utility programs, discussed below.

There are a number of additional output files which you might find useful. These are discussed below.

## 6.10 Output Files

The LITTLE\_R program generates several ASCII text files to detail the actions taken on observations through a time cycle of the program (sorting, error checking, quality control flags, vertical interpolation). In support of users wishing to plot the observations used for each variable (at each level, at each time), a file is created with this information. Primarily, the ASCII text files are for consumption by the developers for diagnostic purposes. The main output of the LITTLE\_R program is the gridded, pressure-level data set to be passed to the INTERPF program (file LITTLE\_R\_DOMAIN<#>).

In each of the files listed below, the text “\_YYYY-MM-DD\_HH:mm:ss.tttt” allows each time period that is processed by LITTLE\_R to output a separate file. The only unusual information in the date string is the final four letters “tttt” which is the decimal time to ten thousandths (!) of a second. The bracketed “[sfc\_fdda\_]” indicates that the surface FDDA option of LITTLE\_R creates the same set of files with the string “sfc\_fdda\_” inserted.

### 6.10.1 LITTLE\_R\_DOMAIN<#>

The final analysis file at surface and pressure levels. Generating this file is the primary goal of running LITTLE\_R.

### 6.10.2 SFCFDDA\_DOMAIN<#>

Use of the surface FDDA option in LITTLE\_R creates a file called “SFCFDDA\_DOMAIN<#>”. This file contains the surface analyses at INTF4D intervals, analyses of T, u, v, RH,  $q_v$ ,  $p_{sfc}$ ,  $p_{msl}$ , and a count of observations within 250 km of each grid point.

### 6.10.3 result\_out\_[sfc\_fdda\_]YYYY-MM-DD\_HH:mm:ss.tttt

This file contains a listing of all of the observations available for use by the LITTLE\_R program. The observations have been sorted and the duplicates have been removed. Observations outside of the analysis region have been removed. Observations with no information have been removed. All reports for each separate location (different levels but at the same time) have been combined to form a single report. Interspersed with the output data are lines to separate each report. This file contains reports discarded for QC reasons.

### 6.10.4 useful\_out\_[sfc\_fdda\_]YYYY-MM-DD\_HH:mm:ss.tttt

This file contains a listing of all of the observations available for use by the LITTLE\_R program. The observations have been sorted and the duplicates have been removed. Observations outside of the analysis region have been removed. Observations with no information have been removed. All reports for each separate location (different levels but at the same time) have been combined to form a single report. Data which has had the “discard” flag internally set (data which will not be sent to the quality control or objective analysis portions of the code) are not listed in this output. No additional lines are introduced to the output, allowing this file to be reused as an observation input file.



### 6.10.5 discard\_out\_[sfc\_fdda\_]YYYY-MM-DD\_HH:mm:ss.tttt

This file only contains a listing of the discarded reports. This is a good place to begin to search to determine why an observation didn't make it into the analysis. This file has additional lines interspersed within the output to separate each report.

### 6.10.6 qc\_out\_[sfc\_fdda\_]YYYY-MM-DD\_HH:mm:ss.tttt

The information contained in the qc\_out file is similar to the useful\_out. The data has gone through a more expensive test to determine if the report is within the analysis region, and the data have been given various quality control flags. Unless a blatant error in the data is detected (such as a negative sea-level pressure), the observation data are not typically modified, but only assigned quality control flags. Any data failing the error maximum or buddy check tests are not used in the objective analysis.

### 6.10.7 obs\_used\_for\_oa\_out\_[sfc\_fdda\_]YYYY-MM-DD\_HH:mm:ss.tttt

This file lists data by variable and by level, where each observation that has gone into the objective analysis is grouped with all of the associated observations for plotting or some other diagnostic purpose. The first line of this file is the necessary FORTRAN format required to input the data. There are titles over the data columns to aid in the information identification. Below are a few lines from a typical file.

```
( 3x,a8,3x,i6,3x,i5,3x,a8,3x,2(g13.6,3x),2(f7.2,3x),i7 )
Number of Observations 00001214
```

Variable Name	Press Level	Obs Number	Station ID	Obs Value	Obs-1st Guess	X Location	Y Location	QC Value
U	1001	1	CYYT	6.39806	4.67690	161.51	122.96	0
U	1001	2	CWRA	2.04794	0.891641	162.04	120.03	0
U	1001	3	CWVA	1.30433	-1.80660	159.54	125.52	0
U	1001	4	CWAR	1.20569	1.07567	159.53	121.07	0
U	1001	5	CYQX	0.470500	-2.10306	156.58	125.17	0
U	1001	6	CWDO	0.789376	-3.03728	155.34	127.02	0
U	1001	7	CWDS	0.846182	2.14755	157.37	118.95	0

### 6.10.8 plotobs\_out\_[sfc\_fdda\_]YYYY-MM-DD\_HH:mm:ss.tttt

Observations files used by plotting program plot\_level.

## 6.11 Plot Utilities

The LITTLE\_R package provides two utility programs for plotting observations. These programs are called “plot\_soundings” and “plot\_levels”. These optional programs use NCAR Graphics, and are built automatically if the PROGS option in the top-level Makefile is set to \$(I\_HAVE\_NCARG). Both programs prompt the user for additional input options.

### 6.11.1 plot\_soundings

Program plot\_soundings plots soundings. This program generates soundings from either the quality-controlled (“qc\_out\_yyyy-mm-dd\_hh:mm:ss:ffff”) or the non-quality-controlled (“useful\_out\_yyyy-mm-dd\_hh:mm:ss:ffff”) upper air data. Only data that are on the requested analysis levels are processed. The program asks the user for an input filename, and creates the file “soundings.cgm”.

### 6.11.2 plot\_level

Program `plot_level` creates station plots for each analysis level. These plots contain both observations that have passed all QC tests and observations that have failed the QC tests. Observations that have failed the QC tests are plotted in various colors according to which test was failed. The program prompts the user for a date of the form `yyyymmddhh`, and plots the observations from file `"plotobs_out_YYYY-mm-dd_hh:00:00.0000"`. The program creates the file `"levels.cgm"`.

## 6.12 LITTLE\_R Observations Format

To make the best use of the `LITTLE_R` program, it is important for users to understand the `LITTLE_R` Observations Format.

Observations are conceptually organized in terms of reports. A report consists of a single observation or set of observations associated with a single latitude/longitude coordinate.

Examples

- a surface station report including observations of temperature, pressure, humidity, and winds.
- an upper-air station's sounding report with temperature, humidity, and wind observations at many height or pressure levels.
- an aircraft report of temperature at a specific lat/lon/height.
- a satellite-derived wind observation at a specific lat/lon/height.

Each report in the `LITTLE_R` Observations Format consists of at least four records:

- A *report header record*
- one or more *data records*
- an *end data record*
- an *end report record*.

The *report header record* is a 600-character long record (don't worry, much of it is unused and need only dummy values) which contains certain information about the station and the report as a whole: location, station id, station type, station elevation, etc. The report header record is described fully in the following table. Shaded items in the table are unused:

**Report header format**

Variable	Fortran I/O format	Description
latitude	F20.5	station latitude (north positive)
longitude	F20.5	station longitude (east positive)
id	A40	ID of station
name	A40	Name of station
platform	A40	Description of the measurement device
source	A40	GTS, NCAR/ADP, BOGUS, etc.

### Report header format

Variable	Fortran I/O format	Description
elevation	F20.5	station elevation (m)
num_vld_fld	I10	Number of valid fields in the report
num_error	I10	Number of errors encountered during the decoding of this observation
num_warning	I10	Number of warnings encountered during decoding of this observation.
seq_num	I10	Sequence number of this observation
num_dups	I10	Number of duplicates found for this observation
is_sound	L10	T/F Multiple levels or a single level
bogus	L10	T/F bogus report or normal one
discard	L10	T/F Duplicate and discarded (or merged) report.
sut	I10	Seconds since 0000 UTC 1 January 1970
julian	I10	Day of the year
date_char	A20	YYYYMMDDHHmmss
slp, qc	F13.5, I7	Sea-level pressure (Pa) and a QC flag
ref_pres, qc	F13.5, I7	Reference pressure level (for thickness) (Pa) and a QC flag
ground_t, qc	F13.5, I7	Ground Temperature (T) and QC flag
sst, qc	F13.5, I7	Sea-Surface Temperature (K) and QC
psfc, qc	F13.5, I7	Surface pressure (Pa) and QC
precip, qc	F13.5, I7	Precipitation Accumulation and QC
t_max, qc	F13.5, I7	Daily maximum T (K) and QC
t_min, qc	F13.5, I7	Daily minimum T (K) and QC
t_min_night, qc	F13.5, I7	Overnight minimum T (K) and QC
p_tend03, qc	F13.5, I7	3-hour pressure change (Pa) and QC
p_tend24, qc	F13.5, I7	24-hour pressure change (Pa) and QC
cloud_cvr, qc	F13.5, I7	Total cloud cover (oktas) and QC
ceiling, qc	F13.5, I7	Height (m) of cloud base and QC

Following the report header record are the *data records*. These data records contain the observations of pressure, height, temperature, dewpoint, wind speed, and wind direction. There are a number of other fields in the data record that are not used on input. Each data record contains data

for a single level of the report. For report types which have multiple levels (e.g., upper-air station sounding reports), each pressure or height level has its own data record. For report types with a single level (such as surface station reports or a satellite wind observation), the report will have a single data record. The data record contents and format are summarized in the following table

### Format of data records

pressure, qc	F13.5, I7	Pressure (Pa) of observation, and QC
height, qc	F13.5, I7	Height (m MSL) of observation, and QC
temperature, qc	F13.5, I7	Temperature (K) and QC
dew_point, qc	F13.5, I7	Dewpoint (K) and QC
speed, qc	F13.5, I7	Wind speed ( $\text{m s}^{-1}$ ) and QC
direction, qc	F13.5, I7	Wind direction (degrees) and QC
u, qc	F13.5, I7	u component of wind ( $\text{m s}^{-1}$ ), and QC
v, qc	F13.5, I7	v component of wind ( $\text{m s}^{-1}$ ), and QC
rh, qc	F13.5, I7	Relative Humidity (%) and QC
thickness, qc	F13.5, I7	Thickness (m), and QC

The *end data record* is simply a data record with pressure and height fields both set to -777777.

After all the data records and the end data record, an *end report record* must appear. The end report record is simply three integers which really aren't all that important.

### Format of end\_report records

num_vld_fld	I7	Number of valid fields in the report
num_error	I7	Number of errors encountered during the decoding of the report
num_warning	I7	Number of warnings encountered during the decoding the report

#### 6.12.1 QCFlags

In the observations files, most of the meteorological data fields also have space for an additional integer quality-control flag. The quality control values are of the form  $2^n$ , where  $n$  takes on positive integer values. This allows the various quality control flags to be additive yet permits the decomposition of the total sum into constituent components. Following are the current quality control flags that are applied to observations.

```

pressure interpolated from first-guess height      = 2 ** 1 =      2
temperature and dew point both = 0                = 2 ** 4 =     16
wind speed and direction both = 0                 = 2 ** 5 =     32
wind speed negative                               = 2 ** 6 =     64
wind direction < 0 or > 360                       = 2 ** 7 =    128
level vertically interpolated                     = 2 ** 8 =    256
value vertically extrapolated from single level   = 2 ** 9 =    512
sign of temperature reversed                      = 2 ** 10 =   1024
superadiabatic level detected                    = 2 ** 11 =   2048
vertical spike in wind speed or direction         = 2 ** 12 =   4096
convective adjustment applied to temperature field = 2 ** 13 =   8192
no neighboring observations for buddy check       = 2 ** 14 =  16384
-----
fails error maximum test                         = 2 ** 15 =  32768
fails buddy test                                = 2 ** 16 =  65536
observation outside of domain detected by QC      = 2 ** 17 = 131072

```

### 6.13 LITTLE\_R Namelist

The LITTLE\_R namelist file is called “namelist.input”, and must be in the directory from which LITTLE\_R is run. The namelist consists of nine namelist records, named “record1” through “record9”, with each having a loosely related area of content. Each namelist record, which extends over several lines in the namelist.input file, begins with “&record<#>” (where <#> is the namelist record number) and ends with a slash “/”.

The data in namelist record1 define the analysis times to process:

```

&record1
  start_year   = 1990
  start_month  = 03
  start_day    = 13
  start_hour   = 00
  end_year     = 1990
  end_month    = 03
  end_day      = 14
  end_hour     = 00
  interval     = 21600
/

```

#### NAMELIST RECORD1

Namelist Variable	Variable Type	Description
start_year	INTEGER	4-digit year of the starting time to process
start_month	INTEGER	2-digit month of the starting time to process
start_day	INTEGER	2-digit day of the starting time to process
start_hour	INTEGER	2-digit hour of the starting time to process
end_year	INTEGER	4-digit year of the ending time to process

**NAMelist RECORD1**

Namelist Variable	Variable Type	Description
end_month	INTEGER	2-digit month of the ending time to process
end_day	INTEGER	2-digit day of the ending time to process
end_hour	INTEGER	2-digit hour of the ending time to process
interval	INTEGER	time interval (s) between consecutive times to process

The data in record2 define the names of the input files:

```
&record2
  fg_filename      = '../data/REGRID_DOMAIN1'
  obs_filename     = '../data/obs1300'
  obs_filename     = '../data/obs1306'
  obs_filename     = '../data/obs1312'
  sfc_obs_filename = '../data/obs1300'
  sfc_obs_filename = '../data/obs1303'
  sfc_obs_filename = '../data/obs1306'
  sfc_obs_filename = '../data/obs1309'
  sfc_obs_filename = '../data/obs1312'
/
```

**NAMelist RECORD2**

Namelist Variable	Variable Type	Description
fg_filename	CHARACTER	file name (may include directory information) of the first-guess fields, there is only a single name
obs_filename	CHARACTER	file name(s) (may include directory information) of the observation files, one required for each time period to run through the objective analysis
sfc_obs_filename	CHARACTER	file name(s) (may include directory information) of the observation files to be used for the surface analyses option (only when .F4D=.TRUE.)

The obs\_filename and sfc\_obs\_filename settings can get confusing, and deserve some additional explanation. Use of these files is related to the times and time interval set in namelist record1, and to the F4D options set in namelist record8. The obs\_filename files are used for the analyses for the full 3D dataset, both at upper-air and the surface. The obs\_filename files should contain all observations, upper-air and surface, to be used for a particular analysis at a particular time. The sfc\_obs\_filename is used *only* when F4D=.TRUE., that is, if surface analyses are being created for surface FDDA nudging. The sfc\_obs\_filenames may be the same files as obs\_filenames, and they should probably contain both surface and upper-air observations. The designation "sfc\_obs\_filename" is not intended to indicate that the observations in the file are at the surface,

but rather that the file is used only for a surface analysis prepared for surface FDDA nudging.

There must be an `obs_filename` listed for each time period for which an objective analysis is desired. Time periods are processed sequentially from the starting date to the ending date by the time interval, all specified in `namelist record1`. For the first time period, the file named first in `obs_filename` is used. For the second time period, the file named second in `obs_filename` is used. This pattern is repeated until all files listed in `obs_filename` have been used. For Subsequent time periods (if any), the first guess is simply passed to the output file without objective analysis.

If the F4D option is selected, the files listed in `sfc_obs_filename` are similarly processed for surface analyses, this time with the time interval as specified by `INTF4D`.

The data in the `record3` concern space allocated within the program for observations. These are values that should not frequently need to be modified:

```
&record3
  max_number_of_obs      = 10000
  fatal_if_exceed_max_obs = .TRUE.
/
```

### NAMELIST RECORD3

Namelist Variable	Variable Type	Description
<code>max_number_of_obs</code>	INTEGER	anticipated maximum number of reports per time period
<code>fatal_if_exceed_max_obs</code>	LOGICAL	T/F flag allows the user to decide the severity of not having enough space to store all of the available observations

The data in `record4` set the quality control options. There are four specific tests that may be activated by the user:

```
&record4
  qc_test_error_max      = .TRUE.
  qc_test_buddy          = .TRUE.
  qc_test_vert_consistency = .FALSE.
  qc_test_convective_adj  = .FALSE.
  max_error_t            = 8
  max_error_uv           = 10
  max_error_z            = 16
  max_error_rh           = 40
  max_error_p            = 400
  max_buddy_t            = 10
  max_buddy_uv           = 12
  max_buddy_z            = 16
```

```
max_buddy_rh      = 40
max_buddy_p       = 400
buddy_weight      = 1.0
max_p_extend_t    = 1300
max_p_extend_w    = 1300
/
```

### NAMELIST RECORD4 - QC Options

Namelist Variable	Variable Type	Description
qc_test_error_max	LOGICAL	check the difference between the first-guess and the observation
qc_test_buddy	LOGICAL	check the difference between a single observation and neighboring observations
qc_test_vert_consistency	LOGICAL	check for vertical spikes in temperature, dew point, wind speed and wind direction
qc_test_convective_adj	LOGICAL	remove any super-adiabatic lapse rate in a sounding by conservation of dry static energy

For the error maximum tests, there is a threshold for each variable. These values are scaled for time of day, surface characteristics and vertical level.

### NAMELIST RECORD4 - Error Max Tolerances

Namelist Variable	Variable Type	Description
max_error_t	REAL	maximum allowable temperature difference (K)
max_error_uv	REAL	maximum allowable horizontal wind component difference (m/s)
max_error_z	REAL	not used
max_error_rh	REAL	maximum allowable relative humidity difference (%)
max_error_p	REAL	maximum allowable sea-level pressure difference (Pa)

For the buddy check test, there is a threshold for each variable. These values are similar to stan-



dard deviations.

### NAMELIST RECORD4 - Buddy Check Tolerances

Namelist Variable	Variable Type	Description
max_buddy_t	REAL	maximum allowable temperature difference (K)
max_buddy_uv	REAL	maximum allowable horizontal wind component difference (m/s)
max_buddy_z	REAL	not used
max_buddy_rh	REAL	maximum allowable relative humidity difference (%)
max_buddy_p	REAL	maximum allowable sea-level pressure difference (Pa)
buddy_weight	REAL	value by which the buddy thresholds are scaled

For satellite and aircraft observations, data are often horizontally spaced with only a single vertical level. The following two entries describe how far the user assumes that the data are valid in pressure space.

### NAMELIST RECORD4 - Single Level Extension

Namelist Variable	Variable Type	Description
max_p_extend_t	REAL	pressure difference (Pa) through which a single temperature report may be extended
max_p_extend_w	REAL	pressure difference (Pa) through which a single wind report may be extended

The data in record5 control the enormous amount of print-out which may be produced by the LITTLE\_R program. These values are all logical flags, where TRUE will generate output and FALSE will turn off output.

```
&record5
  print_obs_files      = .TRUE.
  print_found_obs      = .FALSE.
  print_header         = .FALSE.
  print_analysis       = .FALSE.
  print_qc_vert        = .FALSE.
  print_qc_dry         = .FALSE.
  print_error_max      = .FALSE.
  print_buddy          = .FALSE.
  print_oa             = .FALSE.
/
```

The data in record7 concerns the use of the first-guess fields, and surface FDDA analysis options. Always use the first guess.

```
&record7
  use_first_guess      = .TRUE.
  f4d                  = .TRUE.
  intf4d               = 10800
  lagtem               = .FALSE.
/
```

### **NAMELIST RECORD7**

Namelist Variable	Variable Type	Description
use_first_guess	LOGICAL	Always use first guess (use_first_guess=.TRUE.)
f4d	LOGICAL	Turns on (.TRUE.) or off (.FALSE.) the creation of surface analysis files.
intf4d	INTEGER	time interval in seconds between surface analysis times
lagtem	LOGICAL	Use the previous time-period's final surface analysis for this time-period's first guess (lagtem=.TRUE.); or Use a temporal interpolation between upper-air times as the first guess for this surface analysis (lagtem = .FALSE.).

The data in record8 concern the smoothing of the data after the objective analysis. The differences (observation minus first-guess) of the analyzed fields are smoothed, not the full fields:

```
&record8
  smooth_type          = 1
  smooth_sfc_wind      = 1
  smooth_sfc_temp      = 0
  smooth_sfc_rh        = 0
  smooth_sfc_slp       = 0
  smooth_upper_wind    = 0
  smooth_upper_temp    = 0
  smooth_upper_rh      = 0
/
```

## NAMELIST RECORD8

Namelist Variable	Variable Type	Description
smooth_type	INTEGER	1 = five point stencil of 1-2-1 smoothing; 2 = smoother-desmoothing
smooth_sfc_wind	INTEGER	number of smoothing passes for surface winds
smooth_sfc_temp	INTEGER	number of smoothing passes for surface temperature
smooth_sfc_rh	INTEGER	number of smoothing passes for surface relative humidity
smooth_sfc_slp	INTEGER	number of smoothing passes for sea-level pressure
smooth_upper_wind	INTEGER	number of smoothing passes for upper-air winds
smooth_upper_temp	INTEGER	number of smoothing passes for upper-air temperature
smooth_upper_rh	INTEGER	number of smoothing passes for upper-air relative humidity

The data in record9 concern the objective analysis options. There is no user control to select the various Cressman extensions for the radius of influence (circular, elliptical or banana).

```

&record9
  oa_type           = 'MQD'
  mqd_minimum_num_obs = 50
  mqd_maximum_num_obs = 1000
  radius_influence   = 12
  oa_min_switch      = .TRUE.
  oa_max_switch      = .TRUE.
/

```

## RECORD9

Namelist Variable	Variable Type	Description
oa_type	CHARACTER	“MQD” for multiquadric; “Cressman” for the Cressman-type scheme, this string is case dependent
mqd_minimum_num_obs	INTEGER	minimum number of observations for MQD

**RECORD9**

Namelist Variable	Variable Type	Description
mqd_maximum_num_obs	INTEGER	maximum number of observations for MQD
radius_influence	INTEGER	radius of influence in grid units for Cressman scheme
oa_min_switch	LOGICAL	T = switch to Cressman if too few observations for MQD; F = no analysis if too many observations
oa_max_switch	LOGICAL	T = switch to Cressman if too many observations for MQD; F = no analysis if too many observations

**6.14 Fetch.deck**

An IBM job deck is provided to allow users with NCAR IBM access to use the traditional observations archives available to MM5 users from the NCAR Mass Storage System. It is located in the LITTLE\_R/util directory, and called “fetch.deck.ibm”. This job script retrieves the data from the archives for a requested time period, converts it to the LITTLE\_R Observations Format, and stores these reformatted files on the Mass Storage System for the user to retrieve. The critical portion of the script is printed below:

```
# *****
# ***** fetch interactive/batch C shell *****
# *****          NCAR IBM's only          *****
# *****          f90 only          *****
# *****
#
# This shell fetches ADP data from the NCAR MSS system and
# converts it into a format suitable for the little_r
# program. The data are stored on the NCAR MSS.
#
# Three types of data files are created:
# obs:DATE          : Upper-air and surface data used as
#                      input to little_R
# surface_obs_r:DATE : Surface data needed for FDDA in little_r
#                      (if no FDDA will be done, these are not
#                      needed, since they are also contained
#                      in obs:DATE)
# upper-air_obs_r:DATE : Upper-air data (this file is contained
#                      in obs:DATE file, and is not needed for
#                      input to little_r)
#
# This should be the user's case or experiment (used in MSS name).
# This is where the data will be stored on the MSS.
#
set ExpName = MM5V3/TEST # MSS path name for output
```

```
set RetPd      = 365           # MSS retention period in days

#       The only user inputs to the fetch program are the beginning
#       and ending dates of the observations, and a bounding box for the
#       observation search.  These dates are given in YYYYMMDDHH.  The
#       ADP data are global, and include the surface observations and
#       upper-air soundings.  A restrictive bounding box (where
#       possible) reduces the cost substantially.
#
#       Note: No observational data are available prior to 1973, and
#             no or limited surface observations are available
#             prior to 1976.

set starting_date = 1993031300
set ending_date   = 1993031400

set lon_e        = 180
set lon_w        = -180
set lat_s        = -90
set lat_n        = 90

#####
#####                               #####
#####       END OF USER  MODIFICATIONS       #####
#####                               #####
#####
```

